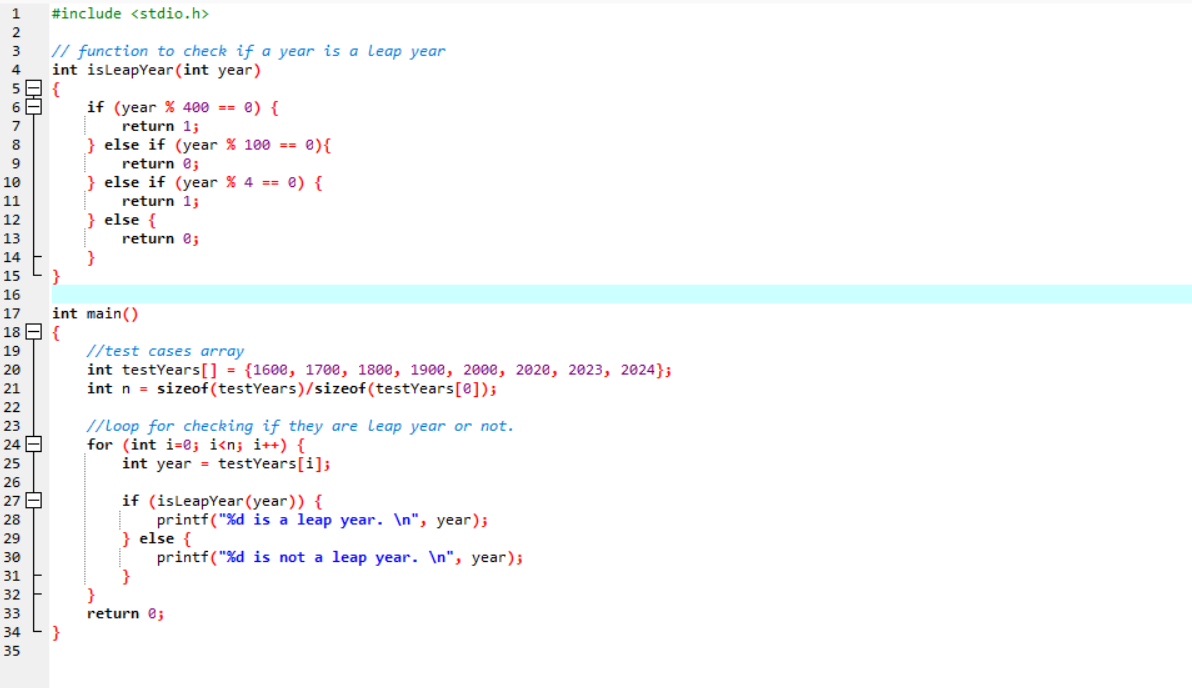
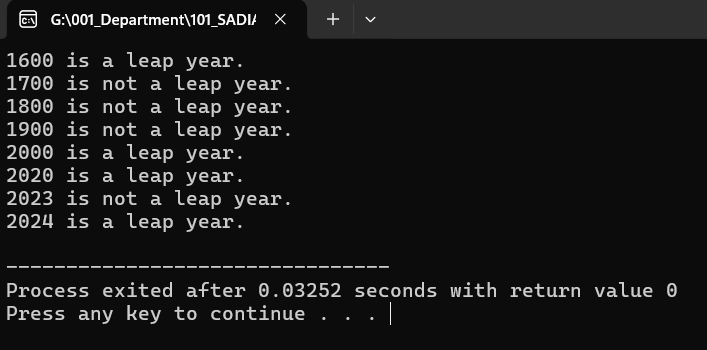
**WEEK – 7**

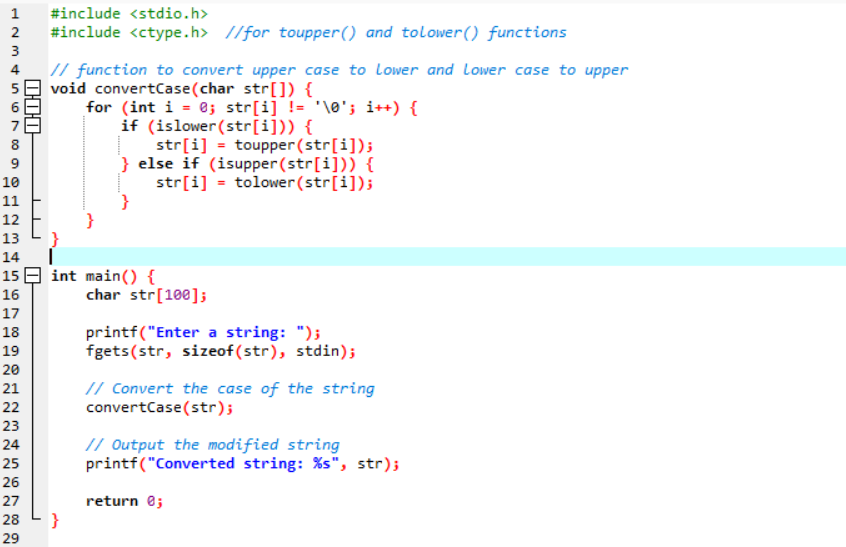
**1****#** C Program to check leap year. All test cases should be present in the in the program.



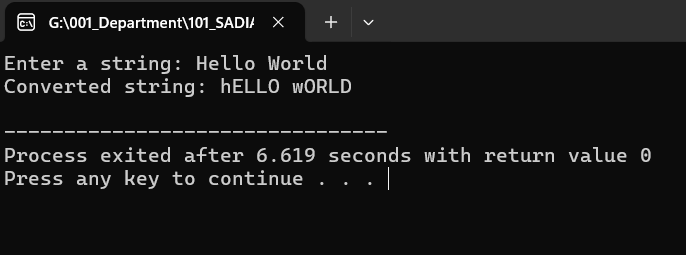
* The isLeapYear function checks if a given year is a leap year based on the rules outlined.
* The main function contains a list of test cases (testYears[]), and the program loops through each year to print whether it is a leap year or not.



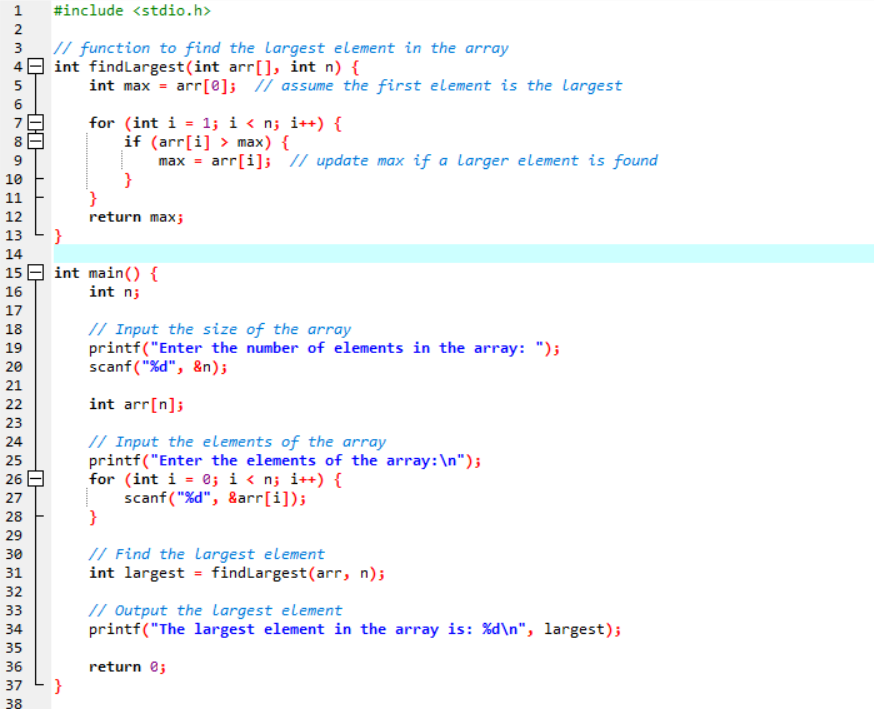
**2#** Program to convert string from upper case to lower case and Lower to upper case.



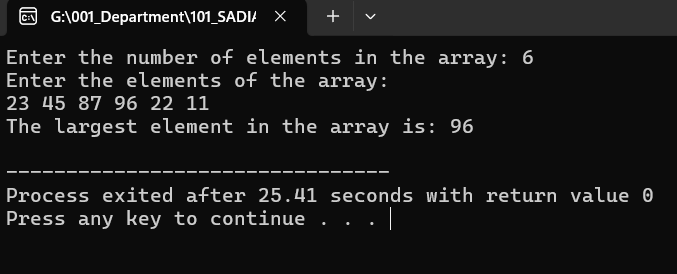
* **islower()** and **isupper()**: These functions check if the character is lowercase or uppercase, respectively.
* **toupper()** and **tolower()**: These functions convert a lowercase letter to uppercase and vice versa.
* **Loop**: The program iterates through each character of the input string and converts it based on whether it is lowercase or uppercase.
* **fgets()**: This is used to read a line of input, allowing spaces in the string.



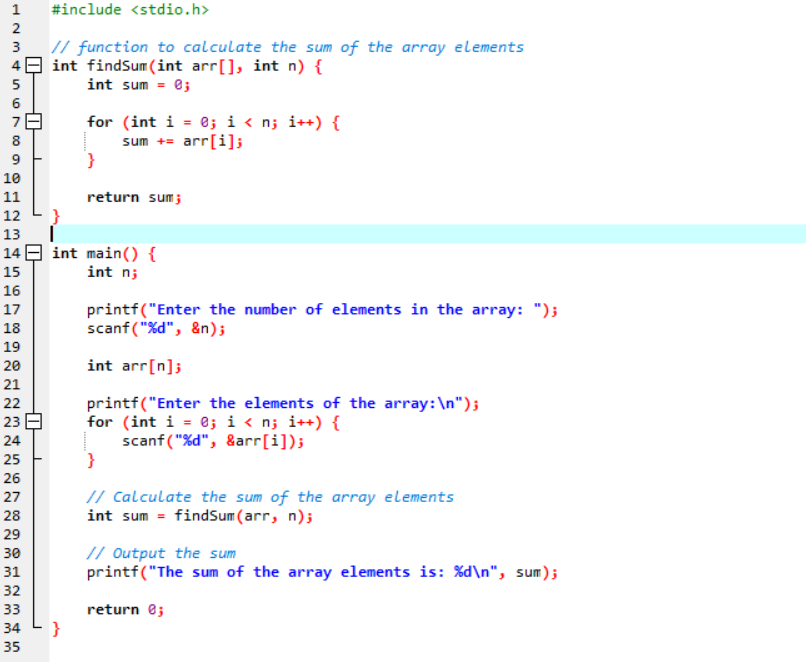
**3#** Find largest element of given array.



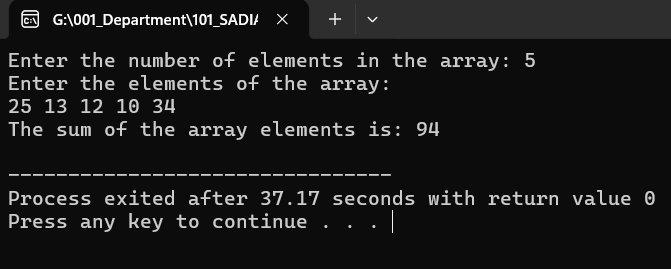
* **findLargest() function**: This function takes the array and its size as input, then iterates through the array to find the largest element. It uses a variable max to store the maximum value, initially assuming the first element is the largest.
* **Array input**: The user inputs the number of elements and then enters the elements of the array.
* The largest element is printed after the function call.



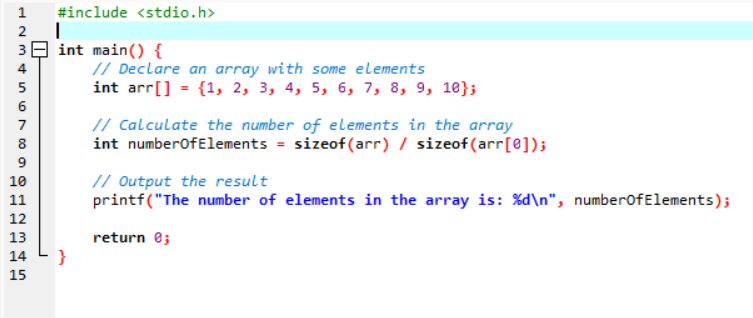
**4#** C program to find sum of array elements.



* **findSum() function**: This function takes the array and its size as input and iterates through the array to calculate the sum by adding each element to a running total (sum).
* **Array input**: The user provides the number of elements (n) and then enters the array elements.
* The result (sum of array elements) is printed after the function call.



**5#** C Program to find number of elements in an array.



* **sizeof(arr)**: This gives the total size of the array in bytes.
* **sizeof(arr[0])**: This gives the size of the first element in the array (since all elements are of the same type).
* **sizeof(arr) / sizeof(arr[0])**: By dividing the total size of the array by the size of one element, you can calculate the number of elements in the array.

